

# Neyman-Pearson (NP) classification algorithms and NP receiver operating characteristic (NP-ROC) curves

**Xin Tong\***

University of Southern California

**Yang Feng<sup>†</sup>**

Columbia University

**Jingyi Jessica Li<sup>‡</sup>**

University of California, Los Angeles

November 21, 2016

## Abstract

In many binary classification applications such as disease diagnosis and spam detection, practitioners often face great needs to control type I errors (e.g., chances of missing a malignant tumor) under some desired threshold. To address such needs, a natural framework is the Neyman-Pearson (NP) classification paradigm, which minimizes population type II errors while installing some upper bound  $\alpha$  on population type I errors. Even though the NP paradigm in hypothesis testing has a century-long history, NP classification has not been studied extensively in the statistics community. Common practices that directly control empirical type I errors under  $\alpha$  do not satisfy the type I error control objective, as the resulting classifiers are likely to have population type I errors much larger than  $\alpha$ . As a result, the NP paradigm has not been properly implemented for many classification scenarios in practice. In this work, we develop the first umbrella algorithm that implements the NP paradigm for popular classification

---

<sup>1</sup>Xin Tong and Yang Feng made equal contribution

<sup>2</sup>To whom correspondence should be addressed. Emails: xint@marshall.usc.edu; jli@stat.ucla.edu

\*Department of Data Sciences and Operations, Marshall School of Business, University of Southern California, Los Angeles, CA, United States

<sup>†</sup>Department of Statistics, Columbia University, New York, NY, United States

<sup>‡</sup>Department of Statistics, University of California, Los Angeles, CA, United States

methods, including logistic regression, support vector machines and random forests. Powered by this umbrella algorithm, we propose a novel evaluation metric for the NP classification methods: the NP receiver operating characteristic (NP-ROC), a variant of the popular receiver operating characteristic (ROC). Despite their conceptual simplicity and wide applicability, ROC curves and their various versions of confidence bands lack information on how to choose a classifier or compare different classifiers whose population type I errors are under some desired threshold with high probability. In contrast, the NP-ROC band will serve as a new effective tool to evaluate, compare and select binary classifiers aiming for population type I error control. We demonstrate the use and properties of the NP umbrella algorithm and the NP-ROC band, available in R package `nproc`, through simulation and real data case studies.

KEYWORDS: receiver operating characteristic curve; classification; Neyman-Pearson paradigm; asymmetric errors; type I error control

In statistics and machine learning, the purpose of classification is to automatically predict discrete outcomes (i.e., class labels) for new observations on the basis of labeled training data. The development of classification theory, methods and applications has been a dynamic area for more than half a century [1]. Well known examples include disease diagnosis, email spam filters, and image classification. The most common type of classification is binary classification, where the class labels are 0 and 1. Most binary classifiers are constructed to optimize the expected classification error: the *risk*, a weighted sum of the *type I error* (the conditional probability of misclassifying a class 0 observation as class 1, also called the *false positive error*) and the *type II error* (the conditional probability of misclassifying a class 1 observation as class 0, also called the *false negative error*), whose weights are marginal probabilities of classes 0 and 1 respectively. We call this setting the *classical classification paradigm* in this paper.

In real-world applications, however, users' priorities for type I and type II errors may differ from these weights. For example, in cancer diagnosis, making a large type I error (i.e., misdiagnosing a cancer patient as healthy) has more severe consequences than making a large type II error (i.e., misdiagnosing a healthy patient with cancer), which may lead to extra medical costs and patient anxiety but will not result in tragic life loss [2, 3, 4, 5]. For such applications, a prioritized control of asymmetric classification errors is in great need. The *Neyman-Pearson (NP) classification paradigm* was developed for this purpose [6, 7, 8, 9], as it seeks

a classifier by minimizing the type II error while controlling the type I error under a user-specified level  $\alpha$ , usually a small value (e.g., 5%). In statistical learning theory, this target classifier is called the oracle NP classifier, which achieves the best type II error given an upper bound  $\alpha$  on its type I error. In contrast to the cost-sensitive learning that also addresses asymmetric classification errors but provides no probabilistic control on the errors, the NP classification paradigm produces classifiers whose type I error is below  $\alpha$  with high probability. Previous works studied the NP classification using both empirical risk minimization (ERM) [6, 10, 11, 7, 12, 8] and plug-in approaches [9, 13]. For a review on the current status of NP classification, we refer the readers to [14].

In this paper, we address two important yet unsolved questions regarding the practicality of the NP classification paradigm and the evaluation of NP classification methods. The first question is how to adapt popular classification methods (e.g., logistic regression [15], support vector machines [16], AdaBoost [17], and random forests [18]) to construct NP classifiers. We address this question by proposing an umbrella algorithm to implement a broad class of classification methods under the NP paradigm. The second question is how to evaluate and compare the performance of different NP classification methods. We propose NP-ROC, a variant of ROC, as a new evaluation tool for NP classification methods.

## 1 Background & mathematical formulation

To facilitate our discussion on technical details, we introduce the following mathematical notations to further explain the classical and the NP classification paradigms. Let  $(X, Y)$  be random variables where  $X \in \mathcal{X} \subset \mathbb{R}^d$  is a vector of  $d$  features and  $Y \in \{0, 1\}$  represents a binary class label. A data set that contains independent observations  $\{(x_i, y_i)\}$  sampled from the joint distribution of  $(X, Y)$  is often divided into training data and test data. Based on training data, a *classifier*  $\phi(\cdot)$  is a mapping  $\phi : \mathcal{X} \rightarrow \{0, 1\}$  that returns the predicted class label given  $X$ . Classification error occurs when  $\phi(X) \neq Y$ , and the binary loss is defined as  $\mathbb{I}(\phi(X) \neq Y)$ , where  $\mathbb{I}(\cdot)$  denotes the indicator function. The *risk* is defined as  $R(\phi) = \mathbb{E}[\mathbb{I}(\phi(X) \neq Y)] = \mathbb{P}(\phi(X) \neq Y)$ , which can be expressed as a weighted sum of type I and II errors:  $R(\phi) = \mathbb{P}(Y = 0)R_0(\phi) + \mathbb{P}(Y = 1)R_1(\phi)$ , where  $R_0(\phi) = \mathbb{P}(\phi(X) \neq Y | Y = 0)$  denotes the (population) *type I error*, and  $R_1(\phi) = \mathbb{P}(\phi(X) \neq Y | Y = 1)$  denotes the (population) *type II error*. The classical classification paradigm aims to mimic the *classical oracle classifier*  $\phi^*$  that minimizes the risk,

$$\phi^* = \arg \min_{\phi} R(\phi).$$

In contrast, the NP classification paradigm aims to mimic the *NP oracle classifier*  $\phi_\alpha^*$  with respect to a pre-specified type I error upper bound  $\alpha$ ,

$$\phi_\alpha^* = \arg \min_{\phi: R_0(\phi) \leq \alpha} R_1(\phi),$$

where  $\alpha$  reflects users' conservative attitude (priority) towards the type I error. Fig. 1 shows a toy example that demonstrates the difference between a classical oracle classifier that minimizes the risk and an NP oracle classifier that minimizes the type II error given type I error  $\leq \alpha = 0.05$ .

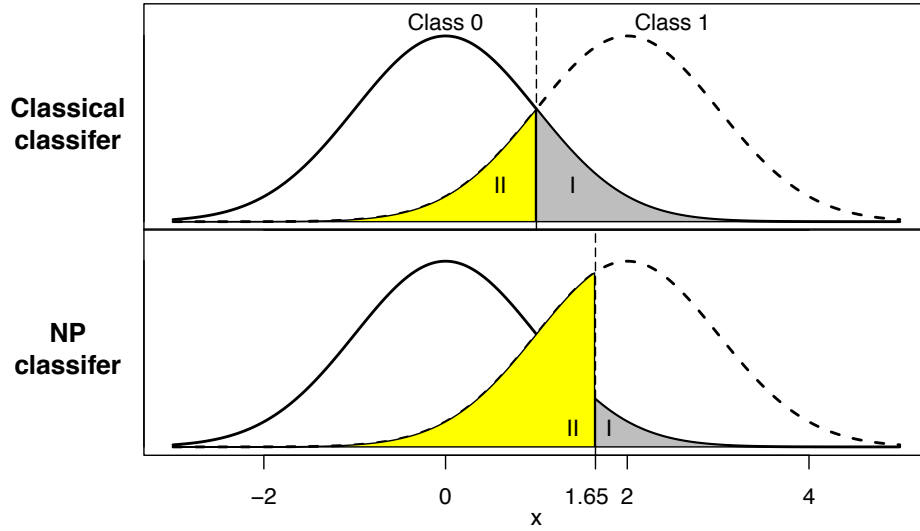


Figure 1: Classical vs. NP oracle classifiers in a binary classification example. The conditional distributions of  $X$  under the two balanced classes are  $\mathcal{N}(0, 1)$  and  $\mathcal{N}(2, 1)$  respectively. Suppose that a user prefers a type I error  $\leq 0.05$ . The classical classifier  $\mathbb{I}(X > 1)$  that minimizes the risk would result in a type I error = 0.16. On the other hand, the NP classifier  $\mathbb{I}(X > 1.65)$  that minimizes the type II error under the type I error constraint ( $\leq 0.05$ ) delivers the desirable type I error.

In practice,  $R(\cdot)$ ,  $R_0(\cdot)$  and  $R_1(\cdot)$  are unobservable because they depend on the unknown joint distribution of  $(X, Y)$ . Instead, their estimates based on data (i.e., the *empirical risk*, *empirical type I error* and *empirical type II error*) are often used in practice. Here we denote the empirical risk and type I and II errors based on training data as  $\hat{R}(\cdot)$ ,  $\hat{R}_0(\cdot)$  and  $\hat{R}_1(\cdot)$  and the empirical risk and type I and II errors based on test data as  $\tilde{R}(\cdot)$ ,  $\tilde{R}_0(\cdot)$  and  $\tilde{R}_1(\cdot)$ .

Because of the wide applications of classification in real-world problems, an army of classification methods have been developed to construct “good” binary classifiers. In this paper, we focus on the *scoring type* of classification methods, which first train a scoring function  $f : \mathcal{X} \rightarrow \mathbb{R}$  using the training data. The scoring function  $f(\cdot)$  assigns a classification score  $f(x)$  to an observation  $x \in \mathbb{R}^d$ . By setting a threshold  $c \in \mathbb{R}$  on the classification scores, a classifier can be obtained. In other words, we consider a classifier  $\phi^c(\cdot) = \mathbb{I}(f(\cdot) > c)$ . Most popular classification methods are of this type [19]. For example, logistic regression, support vector machines, naïve Bayes and neural networks all output a numeric value, i.e., a classification score, to represent the degree to which a test data point belongs to class 1. The classification scores can be strict probabilities or uncalibrated numeric values, as long as a higher score indicates a higher probability of an observation belonging to class 1. Many other classification methods (e.g., random forests) that only output class labels can be converted to the scoring type of classifiers by bagging to generate an ensemble of classifiers, each of which predicts a class label for a test data point, and the proportion of predicted labels being 1 serves as a classification score.

A very popular tool for evaluating classification methods, receiver operating characteristic (ROC) curves are graphical illustration of the overall performance of binary classification methods with all possible type I and type II errors. ROC curves have numerous applications in signal detection theory, diagnostic systems and medical decision making, among other fields [20, 21, 22]. For the scoring type of binary classification methods we focus on in this paper, ROC curves illustrate their overall performance at all possible values of the threshold  $c$  on their output classification scores. An *ROC space* is defined as a two dimensional  $[0, 1] \times [0, 1]$  space, whose horizontal and vertical axes correspond to “type I error” (or “false positive rate”<sup>1</sup>) and “1 - type II error” (or “true positive rate”) respectively. For a binary classification method, its scoring function  $f(\cdot)$  estimated from the training data corresponds to an *ROC curve*, with every point on the curve having the coordinates (type I error, 1 - type II error) corresponding to a threshold  $c$ . The *area under ROC curve* (AUC) is a widely used metric to evaluate a classification method and compare different methods. However, there is much ambiguity in the daily use of ROC analysis. One major issue is the indistinguishable use of the observed empirical type I and II errors and the unobservable population type I and II errors. In practice, typical construction of ROC curves include the following cases: by varying the threshold value  $c$ , points on ROC curves have horizontal and vertical coordinates respectively as

---

<sup>1</sup>In some specific scientific or medical contexts, positiveness and negativeness have strict definitions, and their definition of “false positive rate” may be different from the probability of classifying an observation whose true label is 0 as class 1.

- $\hat{R}_0(\phi^c)$  and  $1 - \hat{R}_1(\phi^c)$  on the training data;
- $\tilde{R}_0(\phi^c)$  and  $1 - \tilde{R}_1(\phi^c)$  on the test data;
- empirical type I error and 1 - empirical type II error estimated by cross validation on the training data.

Besides comparing AUC values, practitioners sometimes construct confidence bands for the ROC curves. Construction methods of ROC bands include those using binomial distribution [23], binormal model [24], bootstrap [25], Working-Hotelling method [24], among others. See [26] for an empirical study of several popular methods.

Although the above mentioned methods of creating ROC curves and bands have been effective to evaluate binary classification methods under the classical paradigm, none of the them is appropriate to evaluate NP classification methods, because they lack information on population type I errors.

## 2 An Umbrella Algorithm for NP Classification

Existing work has proposed several NP classifiers that respect the type I error bound with high probability. They are built upon plug-in density ratio estimates [9] [13]. However, these classifiers are only applicable under a number of restricted scenarios such as low feature dimension [9] and feature independence [13]. Many other statistical and machine learning algorithms have shown as effective classification methods, such as (penalized) logistic regression, support vector machines, AdaBoost, and random forests. To develop NP classifiers that are diverse and adaptive to various practical scenarios, it saves much efforts to implement these popular classification algorithms under the NP paradigm, rather than to construct numerous new NP classifiers based on complex models for density ratios (i.e., class 1 density / class 0 density). The NP classifiers in [9] and [13] have another limitation: the thresholds on density ratios (i.e., the scoring functions of classifiers) are estimated based on concentration inequalities that are universal for all distributions, but could lead to overly conservative (i.e., excessively small) type I errors, in particular for small sample sizes. This is unavoidable as the learning paradigm does not assume knowledge on a specific data distribution. In view of these limitations, we will calculate precise probability regarding order statistics, to find thresholds on classification scores under the NP paradigm.

We propose an umbrella algorithm to adapt popular classification methods to the NP paradigm. Example methods include logistic regression, support vector

machines, AdaBoost, and random forests. Specifically, we seek an efficient way to decide a threshold on the classification scores predicted by each algorithm, so that the thresholds would lead to classifiers with type I errors below the user-specified upper bound  $\alpha$  with high probability. Such an algorithm is in demand because the naïve approach that simply picks a threshold by setting the empirical type I error to  $\alpha$  fails to satisfy the type I error constraint, as demonstrated in the next simulation study.

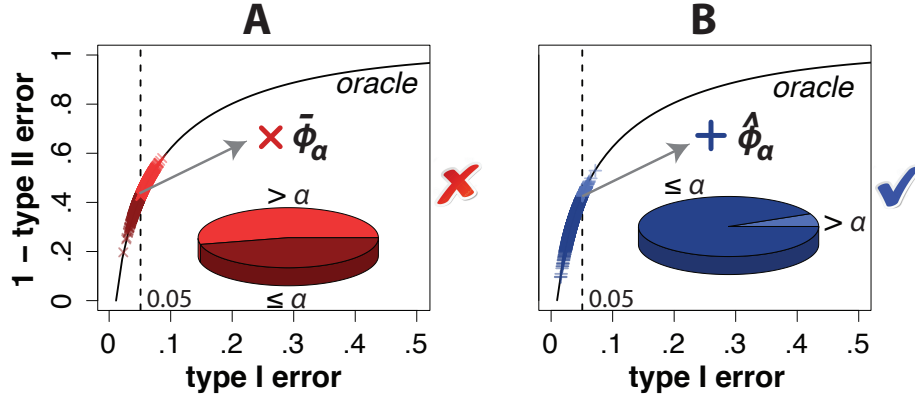


Figure 2: Choose a threshold such that the type I error is below  $\alpha$  with high probability: naïve (A) vs. NP (B) approaches.

## 2.1 Simulation 1

Data are generated from two Gaussian distributions:  $(X|Y = 0) \sim \mathcal{N}(0, 1)$  and  $(X|Y = 1) \sim \mathcal{N}(2, 1)$ , with  $\mathbb{P}(Y = 0) = \mathbb{P}(Y = 1) = 0.5$ . We denote training data from this distribution as  $\{(x_i, y_i)\}_{i=1}^N$ , where  $N = 1,000$ . The classifiers we consider are  $\mathbb{I}(X > c)$ , where  $c \in \mathbb{R}$ . The black curves in Fig. 2 denote the oracle ROC curve, which traces the (population) “type I error” and “1 - type II error” of these classifiers as  $c$  varies. To find a value of  $c$  such that the corresponding classifier has type I error  $\leq \alpha = 0.05$ , a common and intuitive practice is to choose the smallest  $c$  such that the empirical type I error is no greater than  $\alpha$ , resulting in a classifier  $\bar{\phi}_\alpha(\cdot) = \mathbb{I}(\cdot > \bar{c}_\alpha)$ , where  $\bar{c}_\alpha = \inf \left\{ c : \frac{\sum_{i=1}^N \mathbb{I}(x_i > c, y_i = 0)}{\sum_{i=1}^N \mathbb{I}(y_i = 0)} \leq \alpha \right\}$ . In our simulation, we generate  $D = 1,000$  training data sets, and this procedure results in  $D$  classifiers,  $\bar{\phi}_\alpha^{(1)}, \dots, \bar{\phi}_\alpha^{(D)}$ , shown as red “x” on the oracle ROC curve (in Fig. 2, panel A). However, only about half of these classifiers have type I errors

below  $\alpha$ , far from achieving the users' goal. So the common practice does not work well in this case.

In view of this failure, we propose an umbrella algorithm to implement the NP paradigm as pseudocodes in Fig. 3. The essential idea is to choose the smallest threshold on the classification scores such that the violation rate (i.e., the probability that the population type I error exceeds  $\alpha$ ) is controlled under some pre-specified tolerance parameter  $\delta$ . The threshold is to be chosen from an order statistics of classification scores of a left-out class 0 sample, which is not used to train base algorithms (e.g., logistic regression, support vector machines, and random forests). Because we do not impose any assumptions on the underlying data generating process, it is not feasible to establish oracle-type theoretical properties regarding type II errors under this umbrella algorithm. However, since users may favor different base algorithms, this umbrella algorithm is generally applicable given their preferences on type I errors. Before introducing the umbrella algorithm, we first present the following proposition as its theoretical foundation.

**Proposition 1** *Suppose that we divide the training data into two parts: one part with data from both classes 0 and 1 for training a base algorithm (e.g., logistic regression), and another part as a left-out class 0 sample for choosing the classification threshold. Applying the trained base algorithm (scoring function) to the left-out class 0 sample (of size  $n$ ), we denote the resulting classification scores as  $T_1, \dots, T_n$ , which are real-valued random variables. Then we denote by  $T_{(k)}$  the  $k$ -th order statistic (i.e.,  $T_{(1)} \leq \dots \leq T_{(n)}$ ). For a new observation, if we denote its classification score, calculated by the trained base algorithm, as  $T$ , we can construct a classifier  $\hat{\phi}_k = \mathbb{I}(T > T_{(k)})$ . Then the population type I error of  $\hat{\phi}_k$ , denoted by  $R_0(\hat{\phi}_k)$ , is a function of  $T_{(k)}$  and hence a random variable. Assuming the data used to train the base algorithm and the left-out class 0 data are independent, we have*

$$\mathbb{P} \left[ R_0(\hat{\phi}_k) > \alpha \right] \leq \sum_{j=k}^n \binom{n}{j} (1-\alpha)^j \alpha^{n-j}, \quad (1)$$

*that is, the probability that the type I error of  $\hat{\phi}_k$  exceeds  $\alpha$  is under a constant that only depends on  $k$  and  $\alpha$ . We call this probability the violation rate of  $\hat{\phi}_k$ , and denote its upper bound by  $v(k) = \sum_{j=k}^n \binom{n}{j} (1-\alpha)^j \alpha^{n-j}$ . When  $T_i$ 's are continuous, this bound is tight.*

For the proof of Proposition 1, please refer to the Appendix. Note that Proposition 1 is general as it does not rely on any distributional assumptions or on algorithm characteristics.

In view of the above Proposition, if we would like to construct a classifier based on an order statistic of the classification scores of the left-out class 0 sample, the



right order should be

$$k^* = \min \{k \in \{1, \dots, n\} : v(k) \leq \delta\} .$$

It is obvious that  $v(k)$  decreases as  $k$  increases. In order to achieve  $\mathbb{P} \left[ R_0(\hat{\phi}_k) > \alpha \right] \leq \delta$ , that is to control the violation rate under  $\delta$ , at least in the extreme case when  $k = n$ , we need to have the upper bound of the violation rate  $v(n) = (1 - \alpha)^n$  be no larger than  $\delta$ . In other words, we should require  $n \geq \log \delta / \log(1 - \alpha)$ . If the  $n$ -th order statistic cannot guarantee this violation rate control, other order statistics certainly cannot. Therefore for a given  $\alpha$  and  $\delta$ , we need to have the minimum sample size requirement  $n \geq \log \delta / \log(1 - \alpha)$  for type I error violation rate control; otherwise the control cannot be guaranteed, at least by this order statistic approach.

Fig. 3 describes our umbrella NP algorithm. Using this algorithm with  $\alpha = 0.05$ ,  $\delta = 0.05$  and number of random splits  $M = 1$ , we construct  $D$  NP classifiers  $\phi_\alpha^{(1)}, \dots, \phi_\alpha^{(D)}$  based on the  $D = 1,000$  training data sets in Simulation 1. In this simple classification task, the scoring function  $f$  in the algorithm is the identity map. We mark these  $D$  NP classifiers on the oracle ROC curve (shown as blue “+” in Fig. 2 panel B). Unlike the classifiers constructed by the naïve approach in Simulation 1, we see that these  $D$  NP classifiers have type I errors below  $\alpha$  with high probability  $1 - \delta$ .

The following umbrella algorithm includes popular classification methods, such as logistic regression, support vector machine, and random forest, whose detailed implementation can be found in the Appendix. Empirically, multiple random splits  $M > 1$  of training data can be used to increase stability.

### 3 NP-ROC

Here we propose the Neyman-Pearson receiver operating characteristic (NP-ROC) bands as a new evaluation tool for classification methods under the NP paradigm. In the NP-ROC space, the horizontal axis is defined as the type I error (high probability) upper bound, and the vertical axis represents 1 - type II error conditioning on training data. An NP classifier corresponds to a vertical line segment (i.e., a blue dashed line segment in Fig. 4A) in the NP-ROC space. The x-coordinate of a line segment represents the type I error upper bound of that classifier. The y-coordinates of the upper and lower ends of the segment represent the (high probability) upper and lower bounds of (1 - type II error) of that classifier.

To create NP-ROC bands, the sample splitting scheme is slightly different from the umbrella algorithm. We still follow the umbrella algorithm to divide the class 0 data into two halves, using the first half to train the scoring function and the

---

**Algorithm** An NP umbrella algorithm

---

```

1: input:
   training data: a mixed i.i.d. sample  $\mathcal{S} = \mathcal{S}^0 \cup \mathcal{S}^1$ , where  $\mathcal{S}^0$  and  $\mathcal{S}^1$  are class 0 and
   class 1 samples respectively
    $\alpha$ : type I error upper bound,  $0 \leq \alpha \leq 1$ ; [default  $\alpha = 0.05$ ]
    $\delta$ : a small tolerance level,  $0 < \delta < 1$ ; [default  $\delta = 0.05$ ]
    $M$ : number of  $\mathcal{S}^0$  random splits; [default  $M = 1$ ]
2: function RANKTHRESHOLD( $n, \alpha, \delta$ )
3:   for  $k$  in  $\{1, \dots, n\}$  do                                      $\triangleright$  for each rank threshold candidate  $k$ 
4:      $v(k) \leftarrow \sum_{j=k}^n \binom{n}{j} (1-\alpha)^j \alpha^{n-j}$             $\triangleright$  calculate the violation rate with threshold  $k$ 
5:    $k^* \leftarrow \min \{k \in \{1, \dots, n\} : v(k) \leq \delta\}$   $\triangleright$  pick the minimal threshold whose violation rate
   is under  $\delta$ 
6:   return  $k^*$ 
7: procedure NPCLASSIFIER( $\mathcal{S}, \alpha, \delta, M$ )
8:    $n = \lceil |\mathcal{S}^0|/2 \rceil$                                             $\triangleright$  denote half of the size of  $|\mathcal{S}^0|$  as  $n$ 
9:    $k^* \leftarrow \text{RANKTHRESHOLD}(n, \alpha, \delta)$                       $\triangleright$  find the rank threshold
10:  for  $i$  in  $\{1, \dots, M\}$  do                                      $\triangleright$  randomly split  $\mathcal{S}^0$  for  $M$  times
11:     $\mathcal{S}_{i,1}^0, \mathcal{S}_{i,2}^0 \leftarrow \text{random split on } \mathcal{S}^0$            $\triangleright$  each time randomly split  $\mathcal{S}^0$  into two halves with
    equal sizes
12:     $\mathcal{S}_i \leftarrow \mathcal{S}_{i,1}^0 \cup \mathcal{S}^1$                               $\triangleright$  combine  $\mathcal{S}_{i,1}^0$  and  $\mathcal{S}^1$ 
13:     $\mathcal{S}_{i,2}^0 = \{x_1, \dots, x_n\}$                                 $\triangleright$  write  $\mathcal{S}_{i,2}^0$  as a set of  $n$  data points
14:     $f_i \leftarrow \text{classification algorithm}(\mathcal{S}_i)$               $\triangleright$  train a
    classification scoring function  $f_i$  by inputting  $\mathcal{S}_i$  into the classification algorithm; let  $f_i$  output a
    larger expected value for class 1 data
15:     $\mathcal{T}_i = \{t_{i,1}, \dots, t_{i,n}\} \leftarrow \{f_i(x_1), \dots, f_i(x_n)\}$   $\triangleright$  apply the scoring function  $f_i$  to  $\mathcal{S}_{i,2}^0$  to
    obtain a set of score threshold candidates
16:     $\{t_{i,(1)}, \dots, t_{i,(n)}\} \leftarrow \text{sort}(\mathcal{T}_i)$             $\triangleright$  sort elements of  $\mathcal{T}_i$  in an increasing order
17:     $t_i^* \leftarrow t_{i,(k^*)}$   $\triangleright$  find the score threshold corresponding to the chosen rank threshold  $k^*$ 
18:     $\phi_i(X) = \mathbb{I}(f_i(X) > t_i^*)$   $\triangleright$  construct an NP classifier based on the scoring function  $f_i$ 
    and the threshold  $t_i^*$ 
19: output:
   an ensemble NP classifier  $\phi_\alpha(X) = \mathbb{I}\left(\frac{1}{M} \sum_{i=1}^M \phi_i(X) \geq 1/2\right)$   $\triangleright$  by majority vote

```

---

Figure 3: Pseudocode for the NP umbrella algorithm.

second half (size  $n$ ) to estimate the score threshold. However, to estimate the type II error high probability bounds, we need also to divide class 1 data into two halves, using the first half to train the scoring function and the other half to calculate type II error bounds (conditioning on training data). Given a pre-defined tolerance level  $\delta$ , for every possible score threshold among the left-out class 0 scores, we record its rank in an increasing order from 1 to  $n$ . For the  $k$ th order statistic, employ Equation (1) in Proposition 1 to find the  $(1 - \delta)$  probability upper bound  $\alpha$  of  $R_0(\hat{\phi}_k)$  such that  $\alpha = \inf_{\alpha'} \left\{ \mathbb{P} \left[ R_0(\hat{\phi}_k) \leq \alpha' \right] \geq 1 - \delta \right\}$ . We next find the upper and lower bounds of the rank of this score threshold among the left-out class 1 scores as  $r_U$  and  $r_L$  using Equations (2) and (4) in Appendix, and subsequently derive the  $(1 - \delta)$  probability upper and lower bounds  $\beta_U$  and  $\beta_L$  based on Equations (3) and (5) respectively in Appendix. For every rank  $k$ , we calculate  $(\alpha(\hat{\phi}_k), 1 - \beta_U(\hat{\phi}_k))$  and  $(\alpha(\hat{\phi}_k), 1 - \beta_L(\hat{\phi}_k))$  for the classifier  $\hat{\phi}_k$  (shown as the lower and upper ends of blue dashed vertical lines in Fig. 4A). Connecting these points, and varying  $k$  from 1 to  $n$ , we have  $n$  vertical line segments in the NP-ROC space. For a classifier  $\hat{\phi}$  with score thresholds between two ranks of the left-out class 0 scores, say  $k - 1$  and  $k$ , since  $R_0(\hat{\phi}_k) \leq R_0(\hat{\phi}) \leq R_0(\hat{\phi}_{k-1})$  and  $R_1(\hat{\phi}_{k-1}) \leq R_1(\hat{\phi}) \leq R_1(\hat{\phi}_k)$ , we set  $\beta_U(\hat{\phi}) = \beta_U(\hat{\phi}_k)$  and  $\beta_L(\hat{\phi}) = \beta_L(\hat{\phi}_{k-1})$ . Hence we interpolate the  $n$  upper ends of the segments using left continuous step functions, and the  $n$  lower ends using right continuous step functions. The band created after the interpolation is called an NP-ROC band (between the two black curves in Fig. 4A). This band has the interpretation that for every type I error upper bound  $\alpha$ , the achievable (1-type II error) conditioning on training data is sandwiched between the lower and upper ends of the corresponding line segment with chances at least  $1 - 2\delta$ . When we randomly split the training data for  $M > 1$  times and repeat the above procedure, we obtain  $M$  NP-ROC bands. For the  $M$  upper curves and  $M$  lower curves respectively, we calculate the average of vertical values given every horizontal value to obtain an average upper curve and an average lower curve, which form an NP-ROC band for multiple random splits.

In contrast to ROC curves or ROC confidence bands in the literature, NP-ROC bands provide a new perspective for evaluating classification methods: given varying type I error upper bound  $\alpha$ , what are the high probability bounds on the type II error conditioning on training data? NP-ROC bands naturally allow for the comparison of classification methods under the NP paradigm. We use Simulation 2 and Fig. 4B to demonstrate this point.

### 3.1 Simulation 2

Consider the following data generation process.  $(X_1|Y = 0) \sim \mathcal{N}(0, 1)$ ,  $(X_1|Y = 1) \sim \mathcal{N}(1, 1)$ ,  $(X_2|Y = 0) \sim \mathcal{N}(0, 1)$  and  $(X_2|Y = 1) \sim \mathcal{N}(1, 6)$  with  $\mathbb{P}(Y = 0) = \mathbb{P}(Y = 1) = 0.5$ . We simulate one data set with sample size  $N = 1,000$ . We use linear discriminant analysis (LDA) with only  $X_1$  (referred to as method 1) and only  $X_2$  (referred to as method 2). We set the number of data splits  $M = 11$  and the tolerance level  $\delta = 0.1$ . For each classification method, we generate its corresponding NP-ROC band and compare the two bands. The result is presented in Fig. 4B. On top of the horizontal axis, we mark the  $\alpha$  values in black where method 1's lower curve is higher than method 2's upper curve, and in red where method 2's lower curve is higher than method 1's upper curve, so that users can easily decide which method performs better at what  $\alpha$  values under the NP paradigm for a given tolerance level.

One might ask the question: can the popular ROC curves also guide us to compare two classifiers whose type I errors are bounded from the above by some  $\alpha$ , without resorting to the NP umbrella algorithm or the NP-ROC bands? The answer is negative, because again ROC curves are constructed based on empirical type I and type II errors, which are calculated from test data or cross validation on training data, and do not display population type I error information. Concretely, given an ROC curve of a classification method, it is unclear how users should decide which point on the curve corresponds to a classifier satisfying the type I error bound  $\alpha$ . Through Simulation 1 and Fig. 2, we showed that users cannot simply pick the point that has empirical type I error (horizontal axis of the ROC curve) equal to  $\alpha$ , a seemingly intuitive but actually improper practice. We further illustrate this point in Simulation 3 and Fig. 5. Due to the lack of information on population type I error, existing ways of constructing ROC confidence bands cannot serve the purpose either.

### 3.2 Simulation 3

From the same setup as in Simulation 1:  $(X|Y = 0) \sim \mathcal{N}(0, 1)$  and  $(X|Y = 1) \sim \mathcal{N}(2, 1)$ , with  $\mathbb{P}(Y = 0) = \mathbb{P}(Y = 1) = 0.5$ , we simulate  $2D = 2,000$  data sets  $\{(x_i^{(m)}, y_i^{(m)})\}_{i=1}^N$ , where  $m = 1, \dots, 2D$ , and  $N = 1,000$ . The first  $D$  data sets are training data and the rest  $D$  test data. On the  $m$ -th training data set, we construct  $N$  classifiers  $\mathbb{I}(X > x_i^{(m)})$ ,  $i = 1, \dots, N$ , and evaluate their empirical type I and type II errors on the  $m$ -th test data set (i.e., the  $(D + m)$ -th simulated data set), resulting in one ROC curve. We also use the  $m$ -th training data set to calculate one NP-ROC lower curve. Fig. 5 illustrates the  $D = 1,000$  ROC and NP-ROC lower curves.

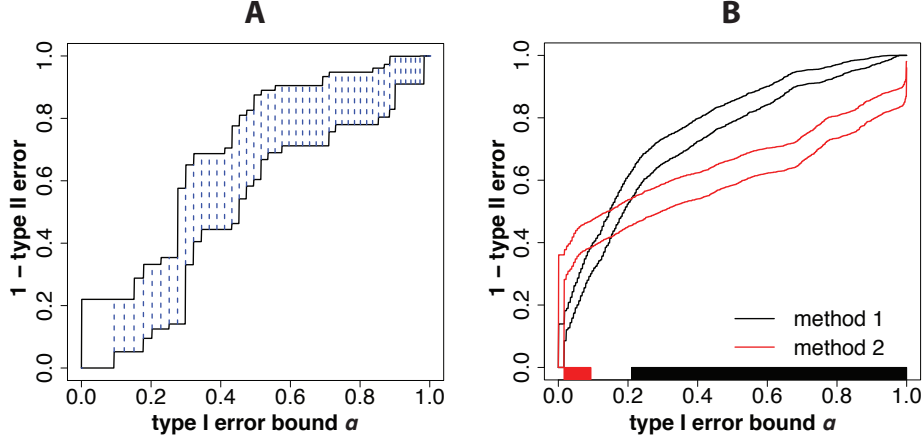


Figure 4: **A:** How to draw an NP-ROC band. Each blue dashed line represents one NP classifier, with horizontal coordinate as  $\alpha$  and vertical coordinates as  $1 - \beta_U$  (lower) and  $1 - \beta_L$  (upper). Left-continuous and right-continuous step functions are used to interpolate points on the upper and lower ends, respectively. **B:** Compare two LDA methods in Simulation 2.

Suppose that users would like to find a classifier respecting a type I error bound  $\alpha = 0.05$  with tolerance level  $\delta = 0.05$  from an ROC curve, an intuitive choice is to pick the classifier at the intersection of the ROC curve and the vertical line at  $\alpha$ . If there is no classifier right at the intersection, the most reasonable way is to pick the first classifier to the left of the intersection. For the  $D$  classifiers chosen in this way, we summarize their empirical type I errors (on the test data) and their population type I errors as histograms (Fig. 5 left panel). The results suggest that although the classifiers have no empirical type I errors greater than  $\alpha$ , their population type I errors have approximately 30% above  $\alpha$ , violating users' desire for controlling type I error under  $\alpha$  with at least 0.95 probability. On the other hand, the NP-ROC lower curves provide a natural way for users to choose classifiers given  $\alpha$ , as the horizontal coordinates of the NP-ROC curves are type I error bounds. Users can simply pick the classifier with horizontal coordinate  $\alpha$ . For the  $D$  chosen NP classifiers, we summarize their empirical type I errors on the test data and their population type I errors as histograms (Fig. 5 right panel). It is clear that the violation rate of population type I error is under  $\delta = 0.05$ .

Another use of the NP-ROC lower curve is that it provides an conservative point-wise estimate of the oracle ROC curve. For an NP classifier  $\hat{\phi}$ , its corresponding point in an NP-ROC lower curve is with high probability in the bottom

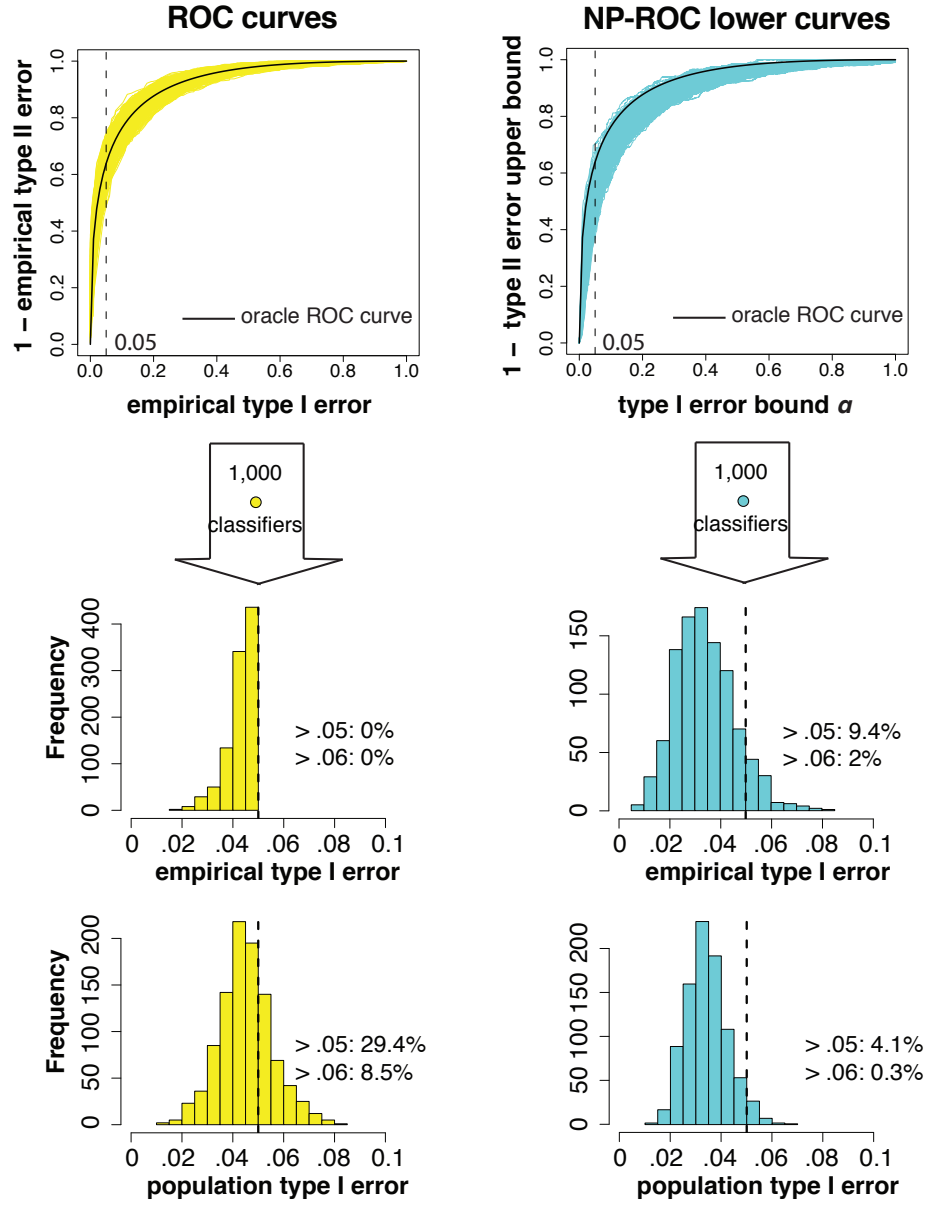


Figure 5: Choose an NP classifier with  $\alpha = .05$  using ROC curves vs. NP-ROC lower curves in Simulation 3.

right corner of its point in the oracle ROC curve. To explain this phenomenon, suppose that the classifier corresponds to point  $(\alpha, 1 - \beta_U)$  in an NP-ROC lower curve and point  $(R_0(\hat{\phi}), 1 - R_1(\hat{\phi}))$  in an oracle ROC curve. Then by the definition of  $\alpha$  and  $\beta_U$  as the high probability upper bound on  $R_0(\hat{\phi})$  and  $R_1(\hat{\phi})$  (conditioning on the training data) respectively, we have  $\alpha \geq R_0(\hat{\phi})$  and  $1 - \beta_U \leq 1 - R_1(\hat{\phi})$  hold with high probability. Hence, the point on the NP-ROC lower curve is to the bottom right of the oracle ROC curve with high probability. In other words, for an NP classifier, its coordinates in the NP-ROC lower curve provide a conservative estimate of its coordinates in the oracle ROC curve. This phenomenon can be visualized in the top right panel of Fig. 5.

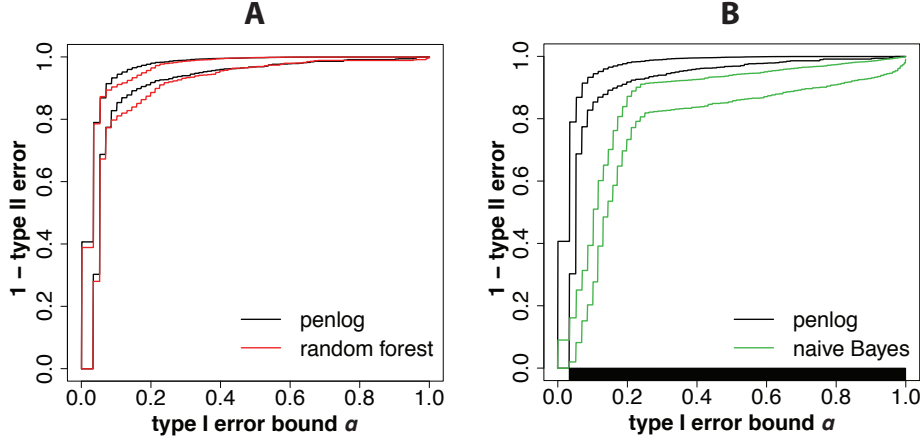


Figure 6: **A:** NP-ROC bands of two classification methods: penalized logistic regression (penlog) and random forests in the real data example, no method dominates the other for any  $\alpha$  values. **B:** The black bar at the bottom indicates the  $\alpha$  values where penlog is better than naïve Bayes with high probability.

## 4 Real Data Example

We implement the umbrella NP algorithm and NP-ROC on a neuroblastoma data set of 43,827 gene expression measurements by Illumina RNA sequencing in 498 neuroblastoma samples (Gene Expression Omnibus Accession number GSE62564), generated by the Sequencing Quality Control (SEQC) Consortium [27, 28, 29, 30]. The gene expression measurements are from GSE62564.SEQC\_NB.RNA-Seq\_log2RPM.txt.gz

available at [31]. Each neuroblastoma sample is labelled as high risk (HR) or non-HR, indicating whether the sample belongs to a high-risk patient by clinical evidence. To predict whether a neuroblastoma sample is HR or not is a binary classification problem, where misclassifying a HR sample as non-HR will have more severe consequences than the other way around. Formulating the problem under the NP classification framework, we consider the HR samples as the class 0 and the non-HR samples as the class 1, each with sample sizes 176 and 322 respectively, and use the 43,827 gene expression measurements as features to classify the samples. Using  $\delta = 0.1$ , we create NP-ROC bands for three classification methods: penalized logistic regression (penlog), random forests and naïve Bayes. In Fig. 6 A, we compare penlog and random forest. At every alpha value, since neither band dominates the other, we declare we cannot distinguish these two methods with confidence across the whole domain. In Fig. 6 B, we compare penlog and naïve Bayes. The long black bar at the bottom of the plot indicates that penlog dominates naïve Bayes for most of the type I upper bound  $\alpha$  values.

## 5 Discussion

In this paper, we propose an umbrella NP algorithm to implement scoring-type classification methods under the NP paradigm. This algorithm guarantees desired high probability control on type I error, allowing us to construct NP classifiers in a wide range of application contexts. We also propose NP-ROC bands, a new variant of the ROC curves under the NP paradigm. NP-ROC bands provide information on the population type I error bounds  $\alpha$  and a range of achievable type II errors for any given  $\alpha$ , so we can use them to compare NP classifiers that aim for any specific type I error bound. Motivated by NP-ROC bands, we will also develop in future works confidence guaranteed ROC bands that treat equally the type I and type II errors.

## 6 Acknowledgements

This work was supported by Zumberge Individual Research Award (Tong), NSF grants DMS-1613338 (Tong, Li), DMS-1308566 (Feng), DMS-1554804 (Feng), Hellman Fellowship (Li), and NIH grant R01GM120507 (Li, Tong).



## References

- [1] Kotsiantis, S. B, Zaharakis, I, & Pintelas, P. (2007) Supervised machine learning: A review of classification techniques. *Informatica* **31**, 249–268.
- [2] Meyer, K. B & Pauker, S. G. (1987) Screening for hiv: can we afford the false positive rate? *New England journal of medicine* **317**, 238–241.
- [3] Liu, S, Babbs, C. F, & Delp, E. J. (2001) Multiresolution detection of spiculated lesions in digital mammograms. *IEEE transactions on Image Processing* **10**, 874–884.
- [4] Dettling, M & Bühlmann, P. (2003) Boosting for tumor classification with gene expression data. *Bioinformatics* **19**, 1061–1069.
- [5] Freeman, E. A & Moisen, G. G. (2008) A comparison of the performance of threshold criteria for binary classification in terms of predicted prevalence and kappa. *Ecological Modelling* **217**, 48–58.
- [6] Cannon, A, Howse, J, Hush, D, & Scovel, C. (2002) Learning with the neyman-pearson and min-max criteria. *Technical Report LA-UR-02-2951*.
- [7] Scott, C & Nowak, R. (2005) A neyman-pearson approach to statistical learning. *IEEE Transactions on Information Theory* **51**, 3806–3819.
- [8] Rigollet, P & Tong, X. (2011) Neyman-pearson classification, convexity and stochastic constraints. *Journal of Machine Learning Research* **12**, 2831–2855.
- [9] Tong, X. (2013) A plug-in approach to nayman-pearson classification. *Journal of Machine Learning Research* **14**, 3011–3040.
- [10] Casasent, D & Chen, X. (2003) Radial basis function neural networks for nonlinear fisher discrimination and neyman-pearson classification. *Neural Networks* **16**, 529 – 535.
- [11] Scott, C. (2005) Comparison and design of neyman-pearson classifiers (Unpublished).
- [12] Han, M, Chen, D, & Sun, Z. (2008) Analysis to Neyman-Pearson classification with convex loss function. *Anal. Theory Appl.* **24**, 18–28.

- [13] Zhao, A, Feng, Y, Wang, L, & Tong, X. (2015) Neyman-Pearson classification under high dimensional settings.
- [14] Tong, X, Feng, Y, & Zhao, A. (2016) A survey on Neyman-Pearson classification and suggestions for future research. *Wiley Interdisciplinary Reviews: Computational Statistics* **8**, 64–81.
- [15] Cox, D. R. (1958) The regression analysis of binary sequences. *Journal of the Royal Statistical Society. Series B (Methodological)* pp. 215–242.
- [16] Cortes, C & Vapnik, V. (1995) Support-vector networks. *Machine learning* **20**, 273–297.
- [17] Freund, Y & Schapire, R. E. (1995) *A decision-theoretic generalization of on-line learning and an application to boosting*. (Springer), pp. 23–37.
- [18] Breiman, L. (2001) Random forests. *Machine learning* **45**, 5–32.
- [19] Fawcett, T. (2006) An introduction to roc analysis. *Pattern recognition letters* **27**, 861–874.
- [20] Hanley, J. A & McNeil, B. J. (1982) The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* **143**, 29–36.
- [21] Bradley, A. P. (1997) The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition* **30**, 1145–1159.
- [22] Pencina, M. J, D’Agostino, R. B, & Vasan, R. S. (2008) Evaluating the added predictive ability of a new marker: from area under the ROC curve to reclassification and beyond. *Statistics in Medicine* **27**, 157–172.
- [23] Hilgers, R. (1991) Distribution-free confidence bounds for roc curves. *Methods of information in medicine* **30**, 96–101.
- [24] Ma, G & Hall, W. (1993) Confidence bands for receiver operating characteristic curves. *Medical Decision Making* **13**, 191–197.
- [25] Campbell, G. (1994) Advances in statistical methodology for the evaluation of diagnostic and laboratory tests. *Statistics in medicine* **13**, 499–508.

- [26] Macskassy, S. A, Provost, F, & Rosset, S. (2005) *ROC confidence bands: An empirical evaluation*. (ACM), pp. 537–544.
- [27] Wang, C, Gong, B, Bushel, P. R, Thierry-Mieg, J, Thierry-Mieg, D, Xu, J, Fang, H, Hong, H, Shen, J, Su, Z, et al. (2014) The concordance between rna-seq and microarray data depends on chemical treatment and transcript abundance. *Nature biotechnology* **32**, 926–932.
- [28] Consortium, S.-I et al. (2014) A comprehensive assessment of rna-seq accuracy, reproducibility and information content by the sequencing quality control consortium. *Nature biotechnology* **32**, 903–914.
- [29] Munro, S. A, Lund, S. P, Pine, P. S, Binder, H, Clevert, D.-A, Conesa, A, Dopazo, J, Fasold, M, Hochreiter, S, Hong, H, et al. (2014) Assessing technical performance in differential gene expression experiments with external spike-in rna control ratio mixtures. *Nature communications* **5**.
- [30] Su, Z, Fang, H, Hong, H, Shi, L, Zhang, W, Zhang, W, Zhang, Y, Dong, Z, Lancashire, L. J, Bessarabova, M, et al. (2014) An investigation of biomarkers derived from legacy microarray data for their utility in the rna-seq era. *Genome biology* **15**, 1–25.
- [31] Su, Z, Shi, L, Fischer, M, & Tong, W. (2014) GSE62564 URL (<http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE62564>). Accessed: 2016-04-25.

## A Properties of Order Statistics

Before proving Proposition 1, we will first introduce the following two lemmas about order statistics.

**Lemma 1** *Let  $T_1, T_2, \dots, T_n$  be independently and identically distributed (i.i.d.) real-valued random variables following a cumulative distribution function (cdf)  $F$ . Denote by  $T_{(k)}$  the  $k$ -th order statistic (i.e.,  $T_{(1)} \leq \dots \leq T_{(n)}$ ). For any  $t$  in the domain of  $T_i$ , we have*

$$\mathbb{P} [T_{(k)} > t] = \sum_{i=n-k+1}^n \binom{n}{i} [1 - F(t)]^i [F(t)]^{n-i}.$$

**Proof** By the property of order statistics,

$$\begin{aligned} & \mathbb{P} [T_{(k)} > t] \\ &= \mathbb{P} [T_{(n)} > t, \dots, T_{(k)} > t] \\ &= \mathbb{P} [\text{at least } (n - k + 1) \text{ of the } T_i\text{'s are greater than } t] \\ &= \sum_{i=n-k+1}^n \mathbb{P} [\text{exactly } i \text{ of the } T_i\text{'s are greater than } t] \\ &= \sum_{i=n-k+1}^n \binom{n}{i} [1 - F(t)]^i [F(t)]^{n-i}. \end{aligned}$$

**Remark:** Lemma 1 does not have any assumptions on  $F$ . Regardless of the continuity of  $F$ , we define its inverse as  $F^{-1}(\cdot) = \inf\{x : F(x) \leq \cdot\}$ , which has a nice property:  $x \leq F(y)$  if and only if  $F^{-1}(x) \leq y$ , for any  $x \in [0, 1]$  and  $y$  in the domain of  $F$ . Letting  $t = F^{-1}(1 - \alpha)$ , we derive Lemma 2 from Lemma 1.

**Lemma 2** *In the same settings as in Lemma 1,*

$$\mathbb{P} [T_{(k)} < F^{-1}(1 - \alpha)] \leq \sum_{j=k}^n \binom{n}{j} (1 - \alpha)^j \alpha^{n-j},$$

*where the equality holds for continuous  $F$ .*

**Proof** Lemma 1 implies that

$$\mathbb{P} [T_{(k)} > F^{-1}(1 - \alpha)] = \sum_{i=n-k+1}^n \binom{n}{i} \alpha^i (1 - \alpha)^{n-i}.$$

It follows that

$$\begin{aligned} & \mathbb{P} [T_{(k)} < F^{-1}(1 - \alpha)] \\ &= \left[ \sum_{i=0}^{n-k} \binom{n}{i} \alpha^i (1 - \alpha)^{n-i} \right] - \mathbb{P} [T_{(k)} = F^{-1}(1 - \alpha)] \\ &= \left[ \sum_{j=k}^n \binom{n}{j} (1 - \alpha)^j \alpha^{n-j} \right] - \mathbb{P} [T_{(k)} = F^{-1}(1 - \alpha)] \\ &\leq \sum_{j=k}^n \binom{n}{j} (1 - \alpha)^j \alpha^{n-j}. \end{aligned}$$

When  $F$  is continuous,  $\mathbb{P} [T_{(k)} = F^{-1}(1 - \alpha)] = 0$  and the above upper bound is tight.

## B Proof of Proposition 1

**Proof** In our context, let  $T_{(k)}$  be the  $k$ -th ordered classification score of a left-out class 0 sample (i.e., class 0 sample not used to train the base algorithms). Suppose  $T_{(k)}$  is the chosen threshold on classification scores. Then the classifier is  $\hat{\phi}_k = \mathbb{I}(T > T_{(k)})$ , where  $T$  is the classification score of an independent observation from class 0, and the population type I error of  $\hat{\phi}_k$  given  $T_{(k)}$  is

$$\mathbb{P} [T > T_{(k)} | T_{(k)}] = 1 - F(T_{(k)}).$$

Then by Lemma 2, the violation rate of using the  $k$ -th ordered classification score as the threshold is

$$\mathbb{P} [1 - F(T_{(k)}) > \alpha] = \mathbb{P} [F(T_{(k)}) < 1 - \alpha] = \mathbb{P} [T_{(k)} < F^{-1}(1 - \alpha)] \leq \sum_{j=k}^n \binom{n}{j} (1 - \alpha)^j \alpha^{n-j},$$

which completes the proof of Proposition 1.

## C Example methods under the umbrella algorithm

Three popular classification methods under the umbrella algorithm are described as follows.

### C.1 Logistic regression

Logistic regression belongs to the class of generalized linear models. It is a popular classification method that models

$$p := \mathbb{P}(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta^T X)}} .$$

for a binary response  $Y$  and feature vector  $X \in \mathbb{R}^d$ . Given training data, logistic regression estimates  $\beta_0$  and  $\beta$  as maximum likelihood estimates  $\hat{\beta}_0$  and  $\hat{\beta}$ . Then given a new observation  $X_{\text{new}}$ , we estimate  $p$  as  $\hat{p}(X_{\text{new}}) = 1 / (1 + e^{-(\hat{\beta}_0 + \hat{\beta}^T X_{\text{new}})})$ . The  $\hat{p}(\cdot)$  can be interpreted as a scoring function. For example, the usual logistic regression  $\mathbb{I}(\hat{p}(\cdot) \geq 1/2)$  threshold  $\hat{p}$  at  $1/2$ . Under the NP paradigm, we potentially need a different choice for  $c$  (other than  $c = 1/2$ ). With each random split in the umbrella algorithm, we can find a threshold  $c$  so that the resulting classifier will have type I error below the desired level  $\alpha$  with high probability. We do multiple random splits for  $\mathcal{S}^0$  (e.g.,  $M = 11$ ) and majority vote to reduce the variance.

### C.2 Support vector machines

Similar to logistic regression, support vector machines (SVM) is also a scoring type of classification method, and its decision boundary is a linear or non-linear function of  $d$  features  $X \in \mathbb{R}^d$ , depending on the kernel it uses. SVM estimates the coefficients of the best separating hyperplane with the largest hard or soft margin in the kernel space of features via solving a quadratic programming problem. For simplicity, we only use the linear kernel here as a representative. Suppose that the estimated separating hyperplane is

$$\hat{f}(X) = X^T \hat{\beta} + \hat{b} = 0 .$$

Then for a new observation  $X_{\text{new}}$ ,  $\hat{f}(X_{\text{new}}) = X_{\text{new}}^T \hat{\beta} + \hat{b}$  can be considered as a SVM classification score, as SVM predicts  $Y_{\text{new}}$  as  $\hat{Y}_{\text{new}} =$

$\mathbb{I}(\hat{f}(X_{\text{new}}) \geq 0)$  under the classical paradigm. By regarding  $\hat{f}(\cdot)$  as a classification scoring function, we use the umbrella algorithm to find a threshold other than 0 on  $\hat{f}(\cdot)$  under the NP paradigm, so that the resulting classifier will have type I error below the desired level  $\alpha$  with high probability. Like the logistic regression, multiple random splits on  $\mathcal{S}^0$  and majority vote can be implemented to reduce variance.

### C.3 Random forests

Random forests (RF) is another popular classification method. Unlike logistic regression and SVM, RF itself is an ensemble of tree-based classifiers, each of which produces a predicted binary label for a new observation. Suppose that the training data have sample size  $n$  and feature dimensionality  $d$ , and RF contain  $N'$  trees. In the training step, every tree in the RF is trained based on a bootstrap sample of size  $n$  from the training data with a random subset of  $l < d$  features. On a new observation  $X_{\text{new}}$ , the  $t$ -th tree will predict a binary label  $\hat{Y}_{\text{new}}^t = \hat{f}^t(X_{\text{new}})$ , and RF will make an overall prediction of  $Y_{\text{new}}$  as  $\hat{Y}_{\text{new}} = \mathbb{I}(\sum_{t=1}^{N'} \hat{f}^t(X_{\text{new}}) > N'/2)$  under the classical paradigm. By regarding  $\hat{f}(\cdot) = \sum_{t=1}^{N'} \hat{f}^t(\cdot)$  as a classification scoring function, each split in the umbrella algorithm leads to a threshold (potentially) other than  $N'/2$  on  $\hat{f}(\cdot)$  under the NP paradigm, and then we can do a majority vote to aggregate the classifiers as a result of the random splits. Therefore, our implementation of the RF under the NP paradigm amounts to create a classifier which is an ensemble of ensembles.

## D Type II error bounds in NP-ROC

Given training data  $\mathcal{S} = \mathcal{S}^0 \cup \mathcal{S}^1$ , where  $\mathcal{S}^0$  and  $\mathcal{S}^1$  are class 0 and class 1 samples respectively. We randomly split  $\mathcal{S}^0$  into  $\mathcal{S}_1^0$  and  $\mathcal{S}_2^0$  and split  $\mathcal{S}^1$  into  $\mathcal{S}_1^1$  and  $\mathcal{S}_2^1$ . For simplicity, we let  $|\mathcal{S}_1^0| = |\mathcal{S}_2^0|$  and  $|\mathcal{S}_1^1| = |\mathcal{S}_2^1|$ , and express the two left-out samples  $\mathcal{S}_2^0$  and  $\mathcal{S}_2^1$  as  $\mathcal{S}_2^0 = \{x_1^0, \dots, x_n^0\}$  and  $\mathcal{S}_2^1 = \{x_1^1, \dots, x_m^1\}$ .

We train a base classification algorithm (e.g., SVM) on  $\mathcal{S}_1^0 \cup \mathcal{S}_1^1$  and denote the resulting classification scoring function, which maps an observation to a class label, as  $f$ .

Suppose that we decide to use the  $k$ -th order score among the left-out class 0 sample,  $f(x_{(k)}^0)$ , as the score threshold. We then construct an NP

classifier  $\hat{\phi}_k$  (based on one sample split) as

$$\hat{\phi}_k(X) = I(f(X) > f(x_{(k)}^0)) .$$

Let  $r_U$  be the following order in the classification scores of the left-out class 1 sample

$$r_U = \min \{r \in \{1, \dots, m\} : f(x_{(r)}^1) \geq f(x_{(k)}^0)\} , \quad (2)$$

and denote its corresponding classifier as

$$\tilde{\phi}_{r_U}(X) = I(f(X) > f(x_{(r_U)}^1)) .$$

Then  $R_1(\tilde{\phi}_{r_U}) \geq R_1(\hat{\phi}_k)$ , that is, the population type II error of  $\tilde{\phi}_{r_U}$  is no less than that of  $\hat{\phi}_k$ , and the equality holds when  $f(x_{(r_U)}^1) = f(x_{(k)}^0)$ .

Denote the  $m$  classification scores of the left-out class 1 sample as  $T_1^1, \dots, T_m^1$ , where  $T_i^1 = f(x_i^1)$ . Then the  $r_U$ -th order statistic is  $T_{(r_U)}^1$ . Denote by  $T^1$  the classification score of a new class 1 observation. The type II error of  $\tilde{\phi}_{r_U}$  given  $T_{(r_U)}^1$  is

$$\mathbb{P}[T^1 \leq T_{(r_U)}^1 | T_{(r_U)}^1] = F_1(T_{(r_U)}^1) ,$$

where  $F_1$  is the CDF of classification scores of class 1 sample.

Denote by  $\beta_U$  the upper bound of the type II error. Then by Lemma 2, the violation rate of using  $T_{(r_U)}^1$  as the threshold is

$$v_{\text{II}}^U(r_U, \beta_U) := \mathbb{P}[F_1(T_{(r_U)}^1) > \beta_U] = \mathbb{P}[T_{(r_U)}^1 > F_1^{-1}(\beta_U)] = 1 - \sum_{j=r_U}^m \binom{m}{j} \beta_U^j (1-\beta_U)^{m-j}$$

if we assume  $F_1$  to be continuous, which holds for most classification algorithms.

If we want to control the violation rate  $v_{\text{II}}^U(r_U, \beta_U)$  under a pre-defined  $\delta$ , we can find the  $(1 - \delta)$  high probability upper bound of type II error as

$$\beta_U^\delta = \inf \left\{ \beta_U \in [0, 1] : \sum_{j=r_U}^m \binom{m}{j} \beta_U^j (1-\beta_U)^{m-j} \geq 1 - \delta \right\} . \quad (3)$$

Since  $R_1(\tilde{\phi}_{r_U}) \leq \beta_U^\delta$  with at least  $(1 - \delta)$  probability, we conclude that  $R_1(\hat{\phi}_k) \leq \beta_U^\delta$  with at least  $(1 - \delta)$  probability.



Similarly, if we want to find a high-probability lower bound  $\beta_L$  on the type II error of  $\hat{\phi}_k$ , we can do the following.

Let  $r_L$  be the following order in the classification scores of the left-out class 1 sample

$$r_L = \max \{r \in \{1, \dots, m\} : f(x_{(r)}^1) \leq f(x_{(k)}^0)\} , \quad (4)$$

and denote its corresponding classifier as

$$\tilde{\phi}_{r_L}(X) = I(f(X) > f(x_{(r_L)}^1)) .$$

Then  $R_1(\tilde{\phi}_{r_L}) \leq R_1(\hat{\phi}_k)$ , that is, the population type II error of  $\tilde{\phi}_{r_L}$  is no greater than  $\hat{\phi}_k$ , and the equality holds when  $f(x_{(r_L)}^1) = f(x_{(k)}^0)$

Then  $f(x_{(r_L)}^1) = T_{(r_L)}^1$ , and the type II error of  $\tilde{\phi}_{r_L}$  given  $T_{(r_L)}^1$  is

$$\mathbb{P}[T^1 \leq T_{(r_L)}^1 | T_{(r_L)}^1] = F_1(T_{(r_L)}^1) .$$

Denote by  $\beta_L$  the lower bound of the type II error. Then by Lemma 2, the violation rate of using  $T_{(r_L)}^1$  as the threshold is

$$v_{\text{II}}^L(r_L, \beta_L) := \mathbb{P}[F_1(T_{(r_L)}^1) < \beta_L] = \mathbb{P}[T_{(r_L)}^1 < F_1^{-1}(\beta_L)] = \sum_{j=r_L}^m \binom{m}{j} \beta_L^j (1-\beta_L)^{m-j} .$$

If we want to control the violation rate  $v_{\text{II}}^L(r_L, \beta_L)$  under a pre-defined  $\delta$ , we can find the  $(1 - \delta)$  high probability lower bound of type II error as

$$\beta_L^\delta = \sup \left\{ \beta_L \in [0, 1] : \sum_{j=r_L}^m \binom{m}{j} \beta_L^j (1-\beta_L)^{m-j} \leq \delta \right\} . \quad (5)$$

Since  $R_1(\tilde{\phi}_{r_L}) \geq \beta_L^\delta$  with at least  $(1 - \delta)$  probability, we conclude that  $R_1(\hat{\phi}_k) \geq \beta_L^\delta$  with at least  $(1 - \delta)$  probability.

Therefore,  $R_1(\hat{\phi}_k) \in [\beta_L^\delta, \beta_U^\delta]$  with probability at least  $1 - 2\delta$ .